

## University of Groningen

### On the design & preservation of software systems

van Gorp, Jilles

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*

Publisher's PDF, also known as Version of record

*Publication date:*

2003

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

van Gorp, J. (2003). *On the design & preservation of software systems*. s.n.

**Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

**Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Rijksuniversiteit Groningen

# ON THE DESIGN & PRESERVATION OF SOFTWARE SYSTEMS

Proefschrift

ter verkrijging van het doctoraat in de  
Wiskunde en Natuurwetenschappen  
aan de Rijksuniversiteit Groningen  
op gezag van de  
Rector Magnificus, dr. F. Zwarts,  
in het openbaar te verdedigen op  
vrijdag 14 februari 2003  
om 16.00 uur

door

Jilles van Gurp  
geboren op 20 oktober 1974  
te Breda

Promotor:

prof. dr. ir. Jan Bosch

Beoordelingscommissie:

prof. dr. S. Demeyer

prof. dr. K. Koskimies

prof. dr. J.C. Van Vliet

# Table of Contents

---

<i>Preface</i> .....	<b>1</b>
<b>CHAPTER 1 Introduction</b> .....	<b>3</b>
1 Software Engineering .....	<b>4</b>
1.1 Software Methodology .....	<b>5</b>
1.2 Components .....	<b>7</b>
1.3 Software Architecture .....	<b>8</b>
1.4 Reuse & Object Orientation.....	<b>9</b>
1.5 Summary .....	<b>10</b>
2 Trends & Motivation .....	<b>11</b>
2.1 More, larger and more complex software .....	<b>11</b>
2.2 Commoditization .....	<b>11</b>
2.3 Variability.....	<b>12</b>
2.4 Erosion .....	<b>12</b>
2.5 Summary .....	<b>13</b>
3 Research Questions.....	<b>13</b>
4 Research Approach .....	<b>14</b>
5 Summary & Remainder of this thesis .....	<b>16</b>
<b>CHAPTER 2 Overview of the Articles</b> .....	<b>17</b>
1 Part One: Object Oriented Frameworks .....	<b>18</b>
2 Part Two: Variability .....	<b>18</b>
3 Part Three: Design erosion .....	<b>18</b>
4 Related publications .....	<b>19</b>

## Part I Object Oriented Frameworks ..... 21

<b>CHAPTER 3 On the implementation of finite state machines</b>	<b>23</b>
1 Introduction .....	<b>23</b>
2 The state pattern .....	<b>24</b>
2.1 FSM Evolution .....	<b>24</b>
2.2 FSM Instantiation .....	<b>26</b>
2.3 Managing Data in a FSM .....	<b>26</b>

3 An Alternative.....	27
3.1 Conceptual Design.....	27
3.2 An Implementation.....	28
4 A Configuration Tool .....	29
4.1 FSMs in XML.....	29
4.2 Configuring and Instantiating.....	30
5 Assessment .....	30
6 Related Work .....	33
7 Conclusion .....	34
<b>CHAPTER 4 Design, Implementation and Evolution of Object Oriented Frameworks: Concepts &amp; Guidelines .....</b>	<b>37</b>
1 INTRODUCTION .....	37
2 THE HAEMO DIALYSIS FRAMEWORK.....	39
3 FRAMEWORK ORGANIZATION .....	41
3.1 Blackbox and Whitebox Frameworks .....	41
3.2 A conceptual model for OO frameworks.....	43
3.3 Dealing with coupling.....	45
3.4 Framework Instantiation .....	46
4 GUIDELINES FOR STRUCURAL IMPROVEMENT .....	47
4.1 The interface of a component should be separated from its implementa- tion .....	47
4.2 Interfaces should be role oriented .....	48
4.3 Role inheritance should be used to combine different roles.....	49
4.4 Prefer loose coupling over delegation .....	50
4.5 Prefer delegation over inheritance .....	51
4.6 Use small components .....	52
5 ADDITIONAL RECOMMENDATIONS.....	53
5.1 Use standard technology .....	54
5.2 Automate configuration.....	55
5.3 Automate documentation .....	55
6 RELATED WORK .....	56
7 CONCLUSION .....	57
7.1 Future Work .....	57
8 Acknowledgements.....	58
<b>CHAPTER 5 Role-Based Component Engineering .....</b>	<b>59</b>
1 Introduction .....	59
1.1 Role-based components .....	59
2 Motivating the use of roles.....	61
3 Role technology .....	64
3.1 Using roles at the design level.....	64
3.2 Using roles at the implementation level.....	65
4 Frameworks and roles.....	66
4.1 Blackbox and white-box frameworks .....	67

4.2 A model for frameworks .....	68
4.3 Dealing with coupling .....	69
5 Summary .....	71
<b>CHAPTER 6 SAABNet: Managing Qualitative Knowledge in Software Architecture Assessment.....</b>	<b>73</b>
1 Introduction.....	73
2 Methodology .....	75
3 SAABNet .....	76
3.1 Qualitative Specification .....	76
3.2 Quantitative Specification.....	79
4 SAABNet usage .....	80
5 Validation .....	81
5.1 Case1: An embedded Architecture.....	82
5.2 Case2: Epoc32 .....	84
6 Related Work .....	87
7 Conclusion.....	88

## Part II Variability ..... 89

<b>CHAPTER 7 On the Notion of Variability in Software Product Lines .....</b>	<b>91</b>
1 Introduction.....	91
1.1 Software Product Lines .....	92
1.2 Problem statement.....	92
1.3 Related work .....	92
1.4 Remainder of the paper .....	94
2 Features: increments of evolution .....	94
3 Variability .....	96
3.1 Abstraction levels.....	97
3.2 Variation point properties.....	98
3.3 Recurring patterns of variability .....	98
4 Case study: Mozilla .....	99
4.1 Mozilla techniques.....	99
4.2 The underlying techniques.....	101
4.3 Summary.....	102
5 Variability management.....	102
6 Conclusion.....	103
<b>CHAPTER 8 A Taxonomy of Variability Realization Techniques .....</b>	<b>105</b>
1 Introduction.....	105
2 Terminology .....	106
3 Introducing Variability in Software Product Lines.....	110

4 Constraining Variability .....	111
4.1 Introducing a Variant Feature .....	112
4.2 Adding of New Variants .....	113
4.3 Binding to a Variant.....	114
4.4 Summary.....	115
5 Variability Realization Techniques.....	116
5.1 Description of the Variability Realization Techniques.....	117
5.2 Summary.....	129
6 Case Studies .....	131
6.1 Axis Communications AB.....	132
6.2 Ericsson Software Technology.....	132
6.3 Mozilla.....	133
6.4 Symbian - Epoc.....	133
6.5 NDC Automation AB.....	133
6.6 Rohill Technologies BV.....	134
6.7 Cummins Inc. ....	134
6.8 Control Channel Toolkit.....	135
6.9 Market Maker .....	135
7 Related Work .....	136
8 Conclusions .....	138

## Part III Design Erosion..... 139

### CHAPTER 9 Design Erosion: Problems & Causes ..... 141

1 Introduction .....	141
1.1 Industrial Examples.....	141
1.2 Problems.....	142
1.3 Optimal vs. minimal approach to Software Development.....	143
1.4 Related work .....	143
1.5 Contributions & Remainder of the paper.....	144
2 The ATM Simulator .....	144
2.1 Version 1: The State Pattern.....	145
2.2 Version 2: The Flyweight Pattern .....	146
2.3 Version 3: Multiple instances + new GUI .....	148
2.4 Version 4: Delegation based approach.....	149
2.5 Version 5: Further decoupling .....	151
2.6 Evolution of ATM Simulator.....	153
3 The minimal strategy .....	155
3.1 Version 1 - 2 .....	155
3.2 Version 2 - 3 .....	156
3.3 Version 3 - 4 .....	156
3.4 Version 4 - 5 .....	156
4 Analysis .....	156
4.1 Architectural drift. ....	156

4.2 Vaporized design decisions .....	157
4.3 Design erosion .....	157
4.4 Accumulated design decisions .....	157
4.5 Limitations of the OO paradigm .....	158
4.6 Optimal vs. minimal strategy .....	158
4.7 Lessons learned .....	158
4.8 Research issues .....	159
4.9 Research Limitations .....	160
5 Conclusion .....	160
5.1 Future work .....	160
<b>CHAPTER 10 Architectural Design Support for Composition &amp; Superimposition .....</b>	<b>163</b>
1 Introduction .....	163
1.1 Problems .....	163
1.2 Running example .....	164
1.3 Solutions .....	164
1.4 Remainder of the paper .....	164
2 Notation .....	165
2.1 Formal Notation .....	166
2.2 Composition .....	168
2.3 Superimposition .....	168
2.4 Interfaces .....	169
2.5 Deriving a detailed design .....	169
3 Examples: The Fire Alarm system .....	170
3.1 Fire alarm with cached sensor deviations .....	172
3.2 Scheduling .....	173
3.3 Blackboard solution .....	173
4 Analysis .....	174
4.1 Problems and Solutions .....	175
4.2 Lessons learned .....	175
4.3 Remaining Issues .....	176
5 Related Work .....	177
6 Conclusion .....	178
6.1 Future Work .....	179
<b>CHAPTER 11 Conclusion .....</b>	<b>181</b>
1 Research Questions .....	182
2 Contributions .....	188
2.1 Part I - Object Oriented Frameworks .....	188
2.2 Part II - Variability .....	189
2.3 Part III - Design Erosion .....	189
3 Future work and open issues .....	190
4 Concluding remarks .....	190
<b>CHAPTER 12 References .....</b>	<b>191</b>



1 Web references .....	<b>197</b>
2 Articles included in this thesis .....	<b>198</b>

# *Preface*

---

After four years of hard work, my Ph. D. thesis is finally finished. I owe some acknowledgements to many people. First of all I would like to thank Jan Bosch who has been my supervisor during the past four years and who has dragged me all the way to Sweden and back. In 1998 when we first met at ECOOP '98, he convinced me to complete my master thesis at his software engineering research group in Ronneby, Sweden. After completing my master thesis there, I continued to work as a Ph. D. student. When he decided to move back to the Netherlands in the summer of 2000, I followed him once again to continue doing research at the University of Groningen.

In addition, I would like to thank the three members of the reading committee, Hans van Vliet, Kai Koskimies and Serge Demeyer for their feedback and for taking the time to read my work. In addition the members of the SEARCH research group in Groningen of which I have been a member, need to be mentioned here: Michel, Eelke, Sybren, Marco, Anton, Rein, Theo-Dirk. Then there are Esmee and Desiree, thanks for assisting me in handling the various bureaucratic aspects of getting a Ph. D. Additionally, I would like to thank the (ex) members of the RISE research group in Blekinge where I completed the first half of my Ph. D as well as my friends and family.

Finally, I would like to thank the various companies (and their representatives) I worked with during the last few years: Axis AB, Ericsson, Symbian, Vertis BV, Rohill Engineering BV, Baan & Thales. Their feedback on my research as well the discussions with their representatives have been invaluable for the work presented in this thesis.

